Agenda

- Operating System Tuning for Informix Database Performance
- OS performance statistics that need to be monitored and collected
- Unix utilities, what to monitor, and scripts to collect data to tune your server
- Performance Metrics Goals

Note: The Examples include here may vary from your OS and version

Operating System Tuning for Informix Database Performance

Start with the Release Notes

- Located in $INFORMIXDIR
  - release/en_us/0333/ids_machine_notes_12.10.txt
- Changes with every release and OS port of Informix

Release Notes

- OS Information
- System Requirements
- System Configuration
- Kernel Parameters
- Feature Notes
- Features not supported

Linux Kernel Shared Memory Calculations

SHMMAX is the maximum size of a single shared memory segment set in bytes.				
kernel.shmmax = 4398046511104				
	Bytes	KB	MB	GB
Recommended	4,398,046,511,104	4,294,967,296	4,194,304	4,096
Default 64Bit Kernel	68,719,476,736	67,108,864	65,536	64
Default 32Bit Kernel	4,294,967,295	4,194,304	4,096	4
Informix Release Notes	33,554,432	32,768	32	0
SHMALL sets the total amount of shared memory pages that can be used system wide, in pages.				
kernel.shmall = 4194304	4194304	17179869184	16777216	16384
SHMALL is the division of SHMMAX/PAGE_SIZE, e.g:. 1073741824/4096=262144.				
	4KB Pages	KB	MB	GB
Recommended	1,073,741,824	4,294,967,296	4,194,304	4,096
Default 64Bit Kernel	16,777,216	67,108,864	65,536	64
Default 32Bit Kernel	1,048,576	4,194,304	4,096	4
Informix Release Notes	4,194,304	16,777,216	16,384	16

Linux Kernel Parameters

- SHMMAX: 4398046511104
- SHMMNI: 4096
- SHMALL: 4194304
- SEMMNI: 4096
- SEMMSL: 250
- SEMMNS: 32000
- SEMOPM: 32

Linux - /etc/sysctl.conf

- # kernel.shmmax = 2147483648
- kernel.shmmax = 4398046511104
- #The maximum number of shared memory segments.
- kernel.shmmni = 4096
- #The maximum amount of shared memory
- kernel.shmall = 4194304
- # semaphores: semmsl, semmns, semopm, semmni
- kernel.sem = 250 128000 100 128

Linux - /etc/sysctl.conf

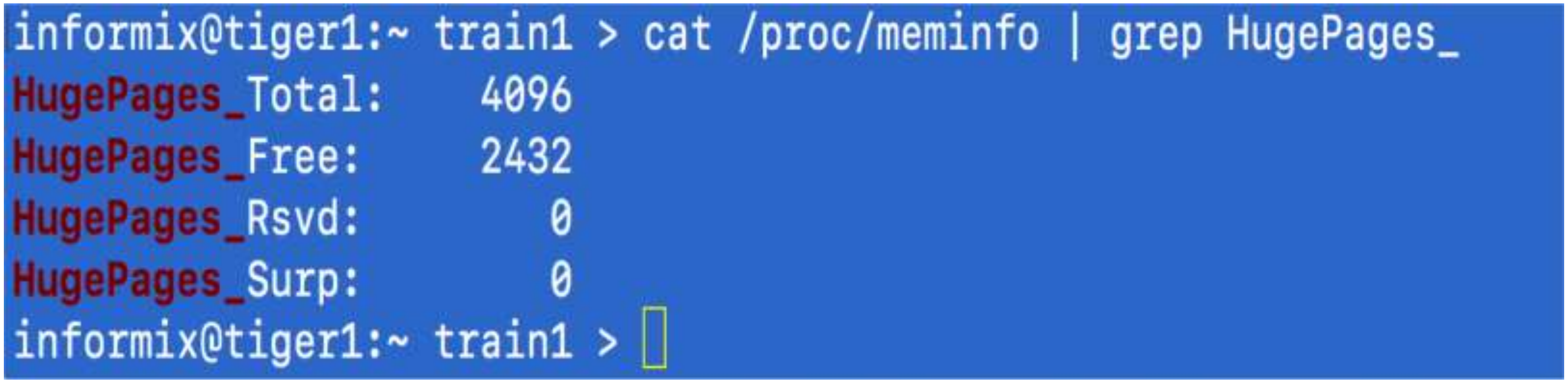
- #How willing linux is to swap memory
- vm.swappiness = 1
- #Filesystem max files increase
- fs.file-max = 2000000
- #Number of active BYTES of allowable concurrent (AIO) requests. Note Informix will only take half of this for its processes, so double what you need. Also if you set AIOON it will only use that number rather then the aio-max-nr setting.
- fs.aio-max-nr = 2048576
- fs.mqueue.msg_max = 1024
- fs.mqueue.queues_max = 4096

Huge Pages

- Informix database server can use large pages in memory when queries
- require more memory
- Message in Online Log on Startup
  - Shared memory segment will use huge pages.

Setting Huge Pages in Linux

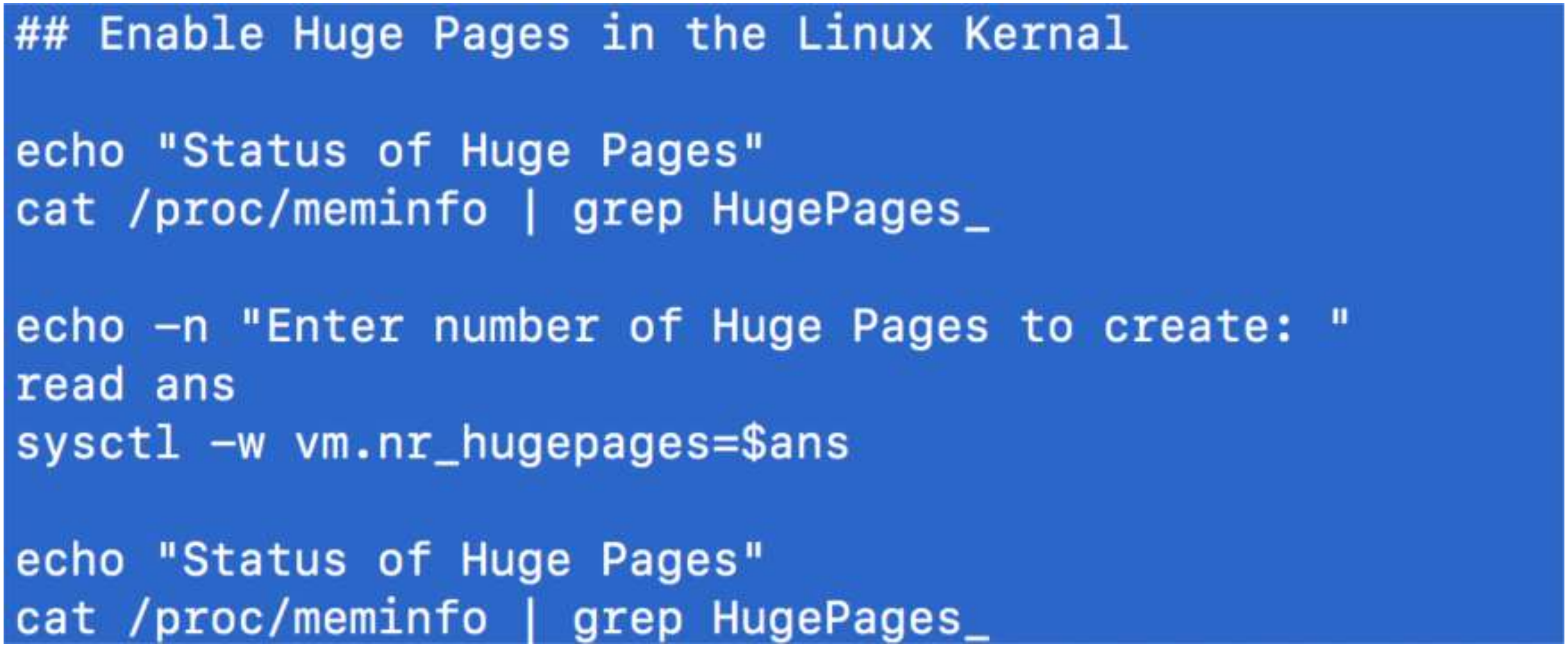
- Cat /proc/meminfo to see if HugePages are used
- Not Used
  - HugePages_Total: = 0
- HugePages Used



```
informix@tiger1:~ train1 > cat /proc/meminfo | grep HugePages_
HugePages_Total:    4096
HugePages_Free:     2432
HugePages_Rsvd:        0
HugePages_Surp:        0
informix@tiger1:~ train1 > 
```

Setting Huge Pages in Linux

- To Set HugePages
  - sysctl -w vm.nr_hugepages=<no. of huge pages>



```
## Enable Huge Pages in the Linux Kernal

echo "Status of Huge Pages"
cat /proc/meminfo | grep HugePages_

echo -n "Enter number of Huge Pages to create: "
read ans
sysctl -w vm.nr_hugepages=$ans

echo "Status of Huge Pages"
cat /proc/meminfo | grep HugePages_
```

Spread the Disk I/O

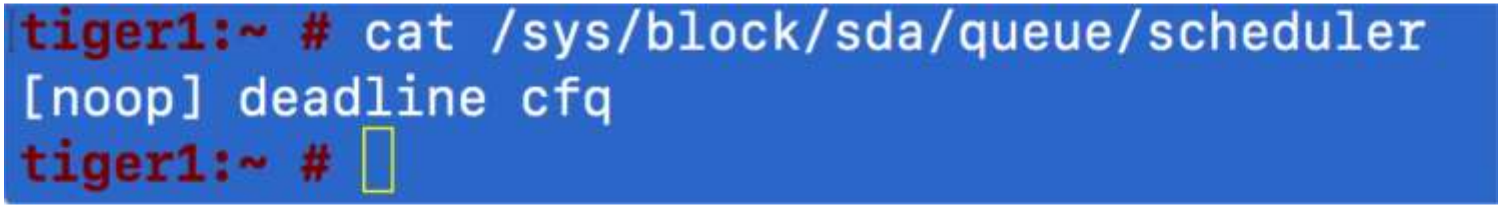
- More disks are better
- Spread the disk I/O across as many disks as possible
- RAID 10 – Stripes and mirrors the data across many disks

Linux Disk I/O Schedular

- noop - recommended for setups with devices that do I/O scheduling themselves, such as intelligent storage or multipathing or databases
- deadline – latency-oriented I/O scheduler
  - default for SSD Drives
- cfg – Completely Fair Queuing
  - default for Spinning Drives

Linux Disk I/O Schedular

- To see the current scheduler
  - cat /sys/block/sda/queue/scheduler
- To Change the current scheduler
  - echo noop > /sys/block/sda/queue/scheduler



```
tiger1:~ # cat /sys/block/sda/queue/scheduler
[noop] deadline cfq
tiger1:~ # 
```

Enable/Disable CPU Hyperthreading on Linux

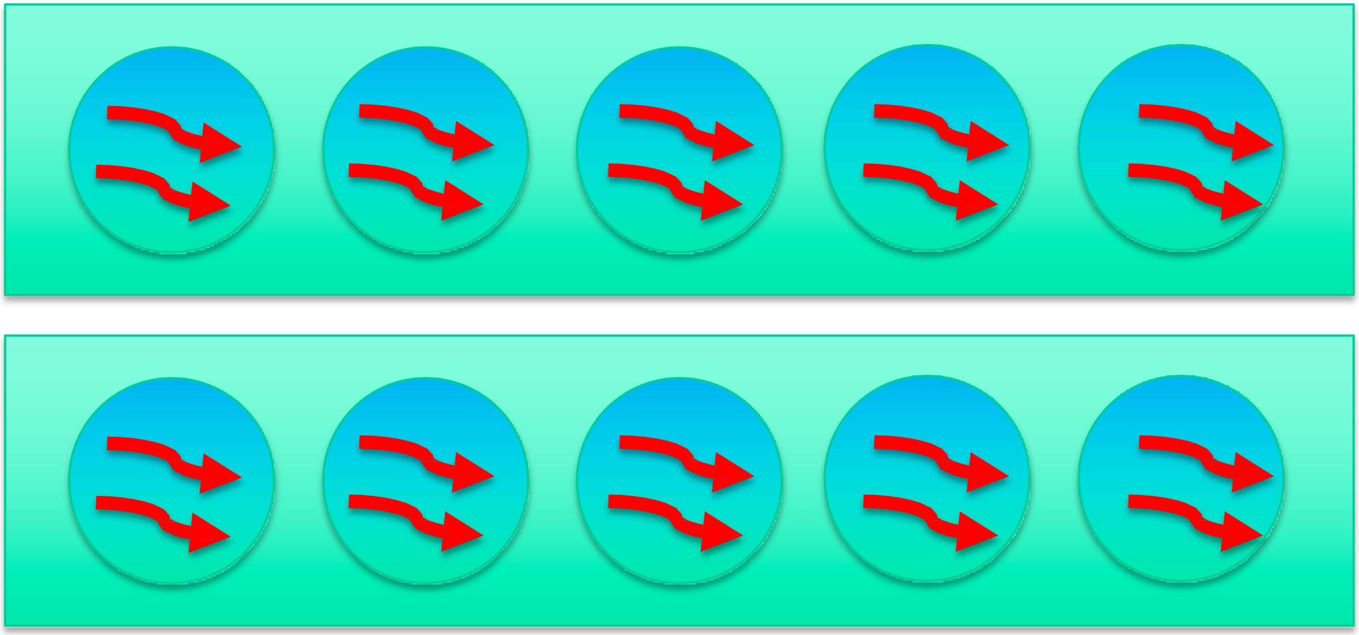
- Best to be done in the BIOS
- To See CPU usage:
  - cat /proc/cpuinfo
  - Look for the CPUs that have the same "core id", you want to switch off the second of each pair

CPU Terms

- Socket = One Chip or Processor
- Cores per Socket = How many cores run on a chip. A core only runs one process at a time.
- Hyper-Threads or SMT threads per Core = Many Cores have the ability to run multiple threads. No matter how many threads run on a Core, only one thread can run at a time on a core. Hyper-Threads will appear as additional Virtual Cores.
- Chip speed is measured in gigahertz (GHz); this is the speed of a single core of your processor.
- PVU - IBM Processor Value Unit = A unit of measure used to differentiate licensing of software

CPU Terms

- Example: 2 Sockets with 5 Cores and 2 Hyper-Threads per Core = 10 Cores and 20 Virtual Cores
- Can run 10 processes at the same time



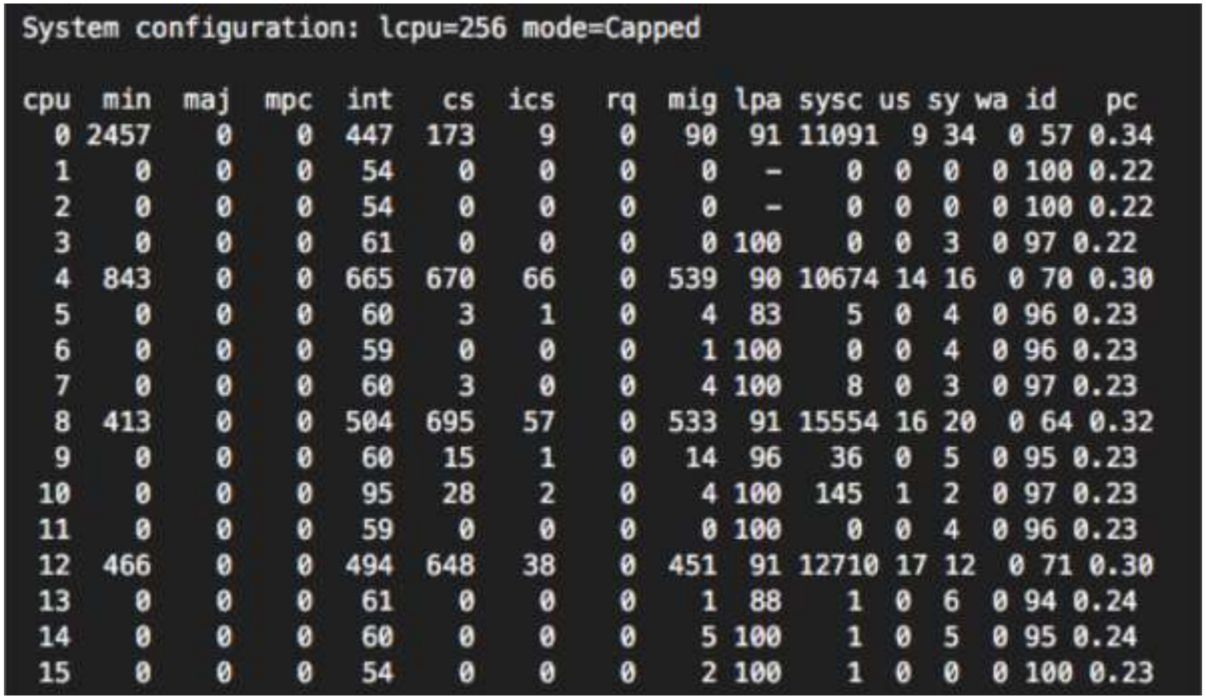
Informix CPU Best Practices

- How many Cores will be allocated for Informix? What else is running on the machine?
- Traditional best practice is number of physical CPU Cores minus 1
- Current CPU Cores are fast enough to handle 2-3 oninits per Core or 1 oninit per 500-1000 MHz

Hyper-Threads or SMT Threads

- Hyper-Threads and SMT Threads may not be helpful to Informix oninit process
- Example:

Informix on AIX. Each AIX Core has 4 SMT threads, each displaying as a CPU; only the first thread is busy, the rest are idle.



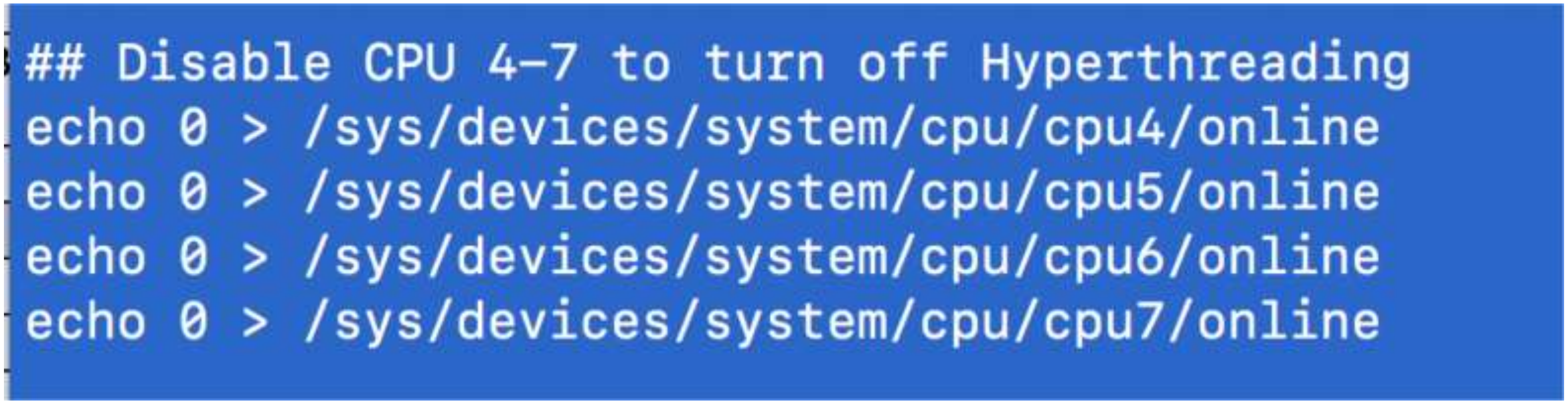
```
System configuration: lcpu=256 mode=Capped

cpu  min  maj  mpc  int   cs  ics   rq  mig lpa  sysc us sy wa  id   pc
  0 2457    0    0  447  173    9    0   90  91 11091  9 34  0  57 0.34
  1    0    0    0   54    0    0    0    0   -     0  0  0  0 100 0.22
  2    0    0    0   54    0    0    0    0   -     0  0  0  0 100 0.22
  3    0    0    0   61    0    0    0    0 100     0  0  3  0  97 0.22
  4  843    0    0  665  670   66    0  539  90 10674 14 16  0  70 0.30
  5    0    0    0   60    3    1    0    4  83     5  0  4  0  96 0.23
  6    0    0    0   59    0    0    0    1 100     0  0  4  0  96 0.23
  7    0    0    0   60    3    0    0    4 100     8  0  3  0  97 0.23
  8  413    0    0  504  695   57    0  533  91 15554 16 20  0  64 0.32
  9    0    0    0   60   15    1    0   14  96    36  0  5  0  95 0.23
 10    0    0    0   95   28    2    0    4 100   145  1  2  0  97 0.23
 11    0    0    0   59    0    0    0    0 100     0  0  4  0  96 0.23
 12  466    0    0  494  648   38    0  451  91 12710 17 12  0  71 0.30
 13    0    0    0   61    0    0    0    1  88     1  0  6  0  94 0.24
 14    0    0    0   60    0    0    0    5 100     1  0  5  0  95 0.24
 15    0    0    0   54    0    0    0    2 100     1  0  0  0 100 0.23
```

Hyper-Threads or SMT Threads Best Practices

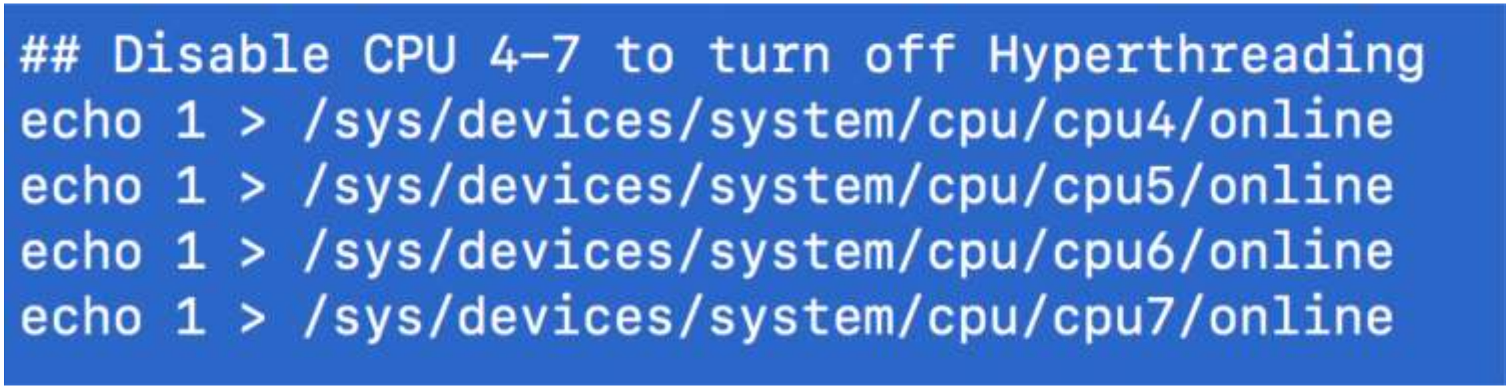
- Test, Test and Test again; don’t assume that more Hyper-Threads or SMT threads are better. Your workload will determine what is best.
- AIX – Try 2 SMT threads per Core on Power6 and Power7, 4 SMT on Power8
- Intel – Try 2 Oninits per Core instead of 2 Hyper-Threads and 1 Oninit per Hyper-Thread

Disable CPU Hyperthreading on Linux



```
## Disable CPU 4-7 to turn off Hyperthreading
echo 0 > /sys/devices/system/cpu/cpu4/online
echo 0 > /sys/devices/system/cpu/cpu5/online
echo 0 > /sys/devices/system/cpu/cpu6/online
echo 0 > /sys/devices/system/cpu/cpu7/online
```

Enable CPU Hyperthreading on Linux



```
## Disable CPU 4-7 to turn off Hyperthreading
echo 1 > /sys/devices/system/cpu/cpu4/online
echo 1 > /sys/devices/system/cpu/cpu5/online
echo 1 > /sys/devices/system/cpu/cpu6/online
echo 1 > /sys/devices/system/cpu/cpu7/online
```

OS performance statistics that need to be monitored and collected

Unix utilities, what to monitor, and scripts to collect data to tune your server

What to Monitor and Tune?

- CPU Usage – How busy are the CPUs?
- Memory Usage – How much memory is being used?
- Disk Usage – What is the disk I/O throughput?
- Network Usage – What is the network utilization?

Tools we will use to Monitor Performance

- SAR – System Activity Recorder
- VMSTAT – CPU and virtual memory statistics
- MPSTAT – Per-CPU statistics
- IOSTAT – Disk I/O throughput statistics
- VXSTAT – Veritas Volume Manager statistics
- PS – Unix processes statistics
- TOP – Top Unix processes statistics
- PSTAT - Top Solaris processes statistics
- NETSTAT – Network statistics

Informix Tools we will use to Monitor Performance

- ONSTAT – Shared memory server statistics
- SYSMASTER DATABASE – Shared memory server statistics
- Server Studio – Command and control center for Informix Server
- OAT – Open Admin Tool for Informix

SAR – System Activity Recorder (old method)

- Setup as a cron job to collect statistics and saves them to a file /var/adm/sa/sa??
- Example Cron setup to collect data every 15 minutes:

```
0,15,30,45 * * * * /usr/lib/sa/sa1
```

- SAR command displays the data collected
- Can also be run in real-time:

```
sar  5 5
```

SAR – System Activity Recorder (New Method)

- Setup via systemctl and runs every 10 minutes:
- To enable SAR

```
systemctl enable sysstat
systemctl start sysstat
systemctl status sysstat -l
```

SAR Reporting Options

-a Report use of file access system routines

-b Report buffer activity

-c Report system calls

-d Report activity for each device (disk or tape drive)

-g Report paging activities

-k Report kernel memory allocation (KMA) activities

-m Report message and semaphore activities

-p Report paging activities

-q Report average queue length

-r Report unused memory pages

-u Report CPU utilization (the default)

-v Report status of process, i-node, file tables

-w Report system swapping and switching activity

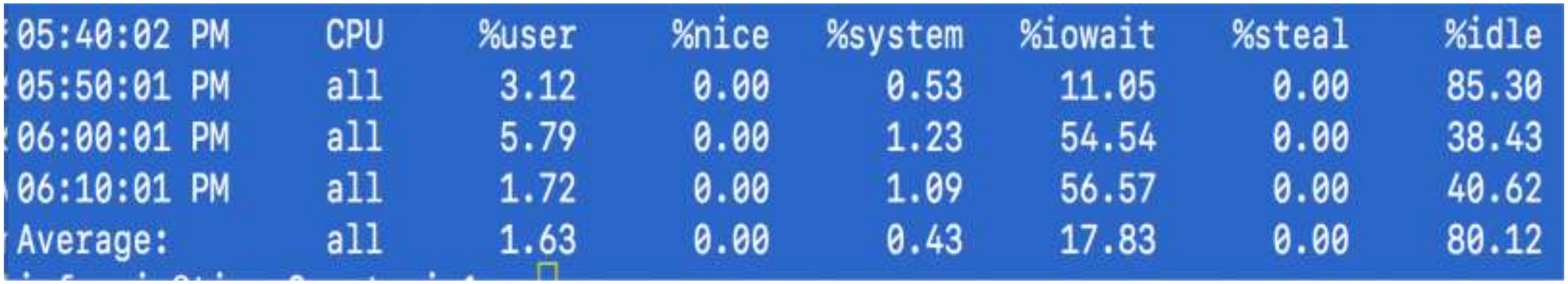
-y Report TTY device activity

-A Report all data. Equivalent to -abcdgkmpqruvwy

SAR Collection Options

- -i sec - Select data at intervals as close as possible to sec seconds.
- -s time - Select data later than time in the form hh[:mm]. Default is 08:00.
- -f filename - Use filename as the data source for sar. Default is the current daily data file /var/log/sa or /var/adm/sa/sadd.
- -o filename - Save samples in file, filename, in binary format.

SAR – Default Output



```
05:40:02 PM     CPU     %user     %nice   %system   %iowait    %steal     %idle
05:50:01 PM     all      3.12      0.00      0.53     11.05      0.00     85.30
06:00:01 PM     all      5.79      0.00      1.23     54.54      0.00     38.43
06:10:01 PM     all      1.72      0.00      1.09     56.57      0.00     40.62
Average:        all      1.63      0.00      0.43     17.83      0.00     80.12
```

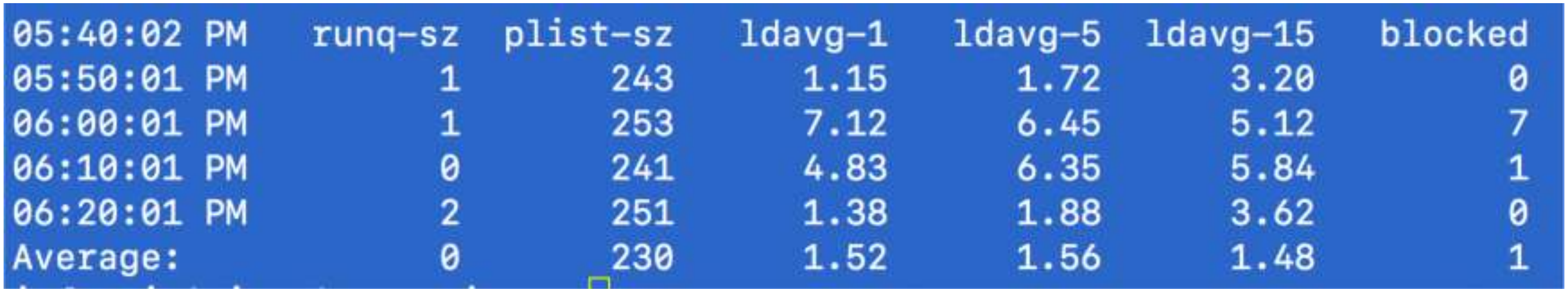
SAR – Default Output

```
00:00:00    %usr    %sys    %wio   %idle
07:00:00    27         3       0      70
07:15:02    61         6       0      33
07:30:01    47         4       0      49
07:45:01    28         3       0      70
08:00:00    30         2       0      68
08:15:00    50         3       0      46
08:30:01    56         3       0      41
08:45:00    22         2       0      77
```

SAR – Default Output

- **Report CPU utilization. The report may show the following fields:**
- %user - Percentage of CPU utilization that occurred while executing at the user level (application).
- %nice - Percentage of CPU utilization that occurred while executing at the user level with nice priority.
- %system - Percentage of CPU utilization that occurred while executing at the system level (kernel). Includes time spent servicing hardware and software interrupts.
- %iowait - Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.
- %steal - Percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor.
- %idle - Percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.

SAR -q (Run Queue)



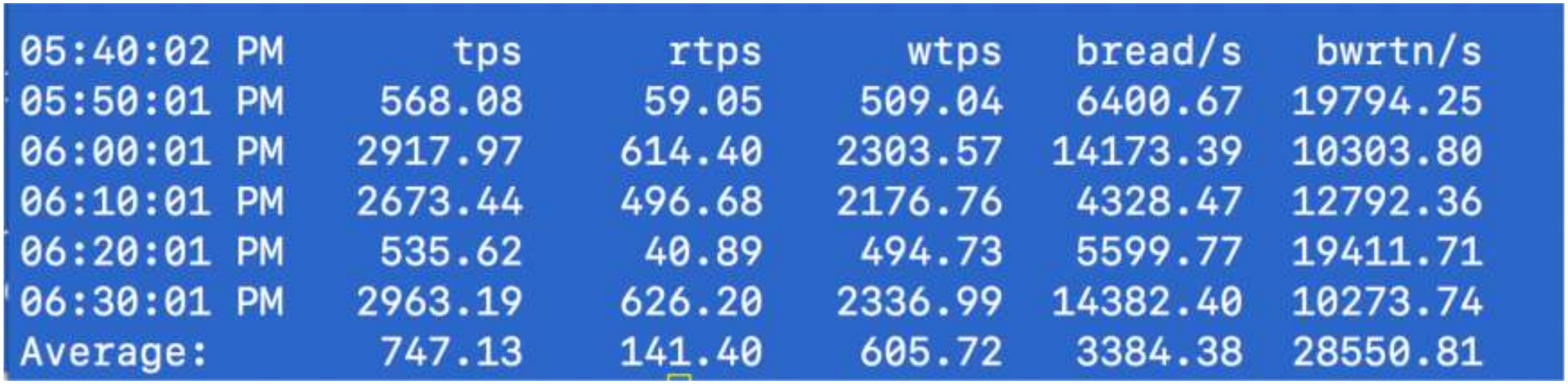
```
05:40:02 PM   runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15   blocked
05:50:01 PM         1       243      1.15      1.72      3.20         0
06:00:01 PM         1       253      7.12      6.45      5.12         7
06:10:01 PM         0       241      4.83      6.35      5.84         1
06:20:01 PM         2       251      1.38      1.88      3.62         0
Average:            0       230      1.52      1.56      1.48         1
```

SAR -q (Run Queue)

•**Report queue length and load averages. The following values are displayed:**

- runq-sz - Run queue length (number of tasks waiting for run time).
- plist-sz - Number of tasks in the task list.
- ldavg-1 - System load average for the last minute. The load average is calculated as the average number of runnable or running tasks (R state), and the number of tasks in uninterruptible sleep (D state) over the specified interval.
- Ldavg-5 - System load average for the past 5 minutes.
- ldavg-15 - System load average for the past 15 minutes.
- blocked - Number of tasks currently blocked, waiting for I/O to complete.

SAR –b (I/O and Transfer Rate Statistics)



```
05:40:02 PM       tps      rtps      wtps   bread/s   bwrtn/s
05:50:01 PM    568.08     59.05    509.04   6400.67  19794.25
06:00:01 PM   2917.97    614.40   2303.57  14173.39  10303.80
06:10:01 PM   2673.44    496.68   2176.76   4328.47  12792.36
06:20:01 PM    535.62     40.89    494.73   5599.77  19411.71
06:30:01 PM   2963.19    626.20   2336.99  14382.40  10273.74
Average:       747.13    141.40    605.72   3384.38  28550.81
```

SAR –b (I/O and Transfer Rate Statistics)

•**Report I/O and transfer rate statistics. The following values are displayed:**

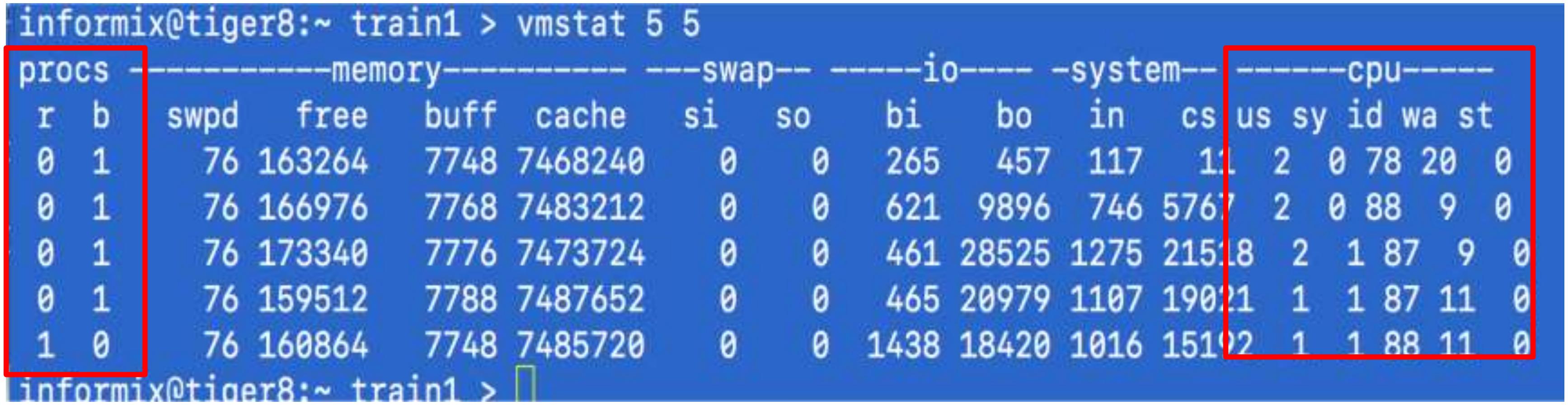
- tps - Total number of transfers per second that were issued to physical devices. A transfer is an I/O request to a physical device. Multiple logical requests can be combined into a single I/O request to the device. A transfer is of indeterminate size.
- rtps - Total number of read requests per second issued to physical devices.
- wtps - Total number of write requests per second issued to physical devices.
- bread/s - Total amount of data read from the devices in blocks per second. Blocks are equivalent to sectors and have a size of 512 bytes.
- bwrtn/s - Total amount of data written to devices in blocks per second.

VMSTAT – CPU and Memory

- Options:

vmstat [-cipsS] [disks] [interval [count]]

- Example:



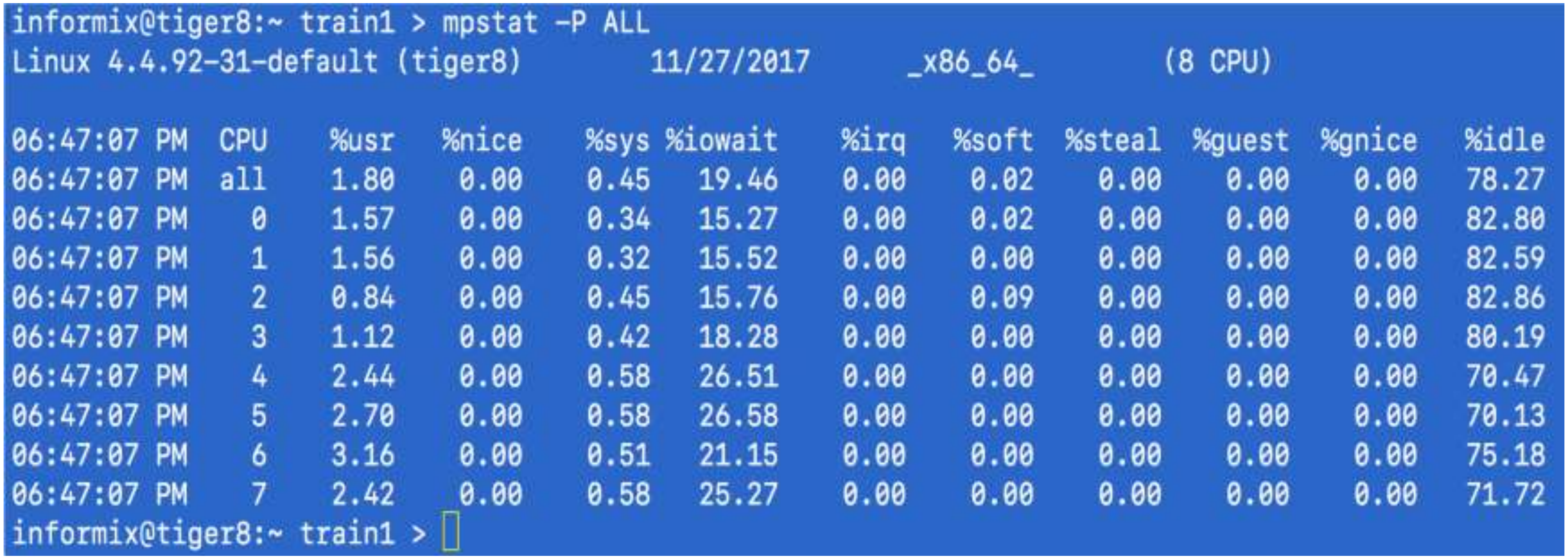
```
informix@tiger8:~ train1 > vmstat 5 5
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  1     76 163264   7748 7468240    0    0   265   457  117   11  2  0 78 20  0
 0  1     76 166976   7768 7483212    0    0   621  9896  746 5767  2  0 88  9  0
 0  1     76 173340   7776 7473724    0    0   461 28525 1275 21518  2  1 87  9  0
 0  1     76 159512   7788 7487652    0    0   465 20979 1107 19021  1  1 87 11  0
 1  0     76 160864   7748 7485720    0    0  1438 18420 1016 15192  1  1 88 11  0
informix@tiger8:~ train1 > 
```

MPSTAT – Per-CPU Statistics

- Options:

mpstat [-p | -P set] [interval [count]]

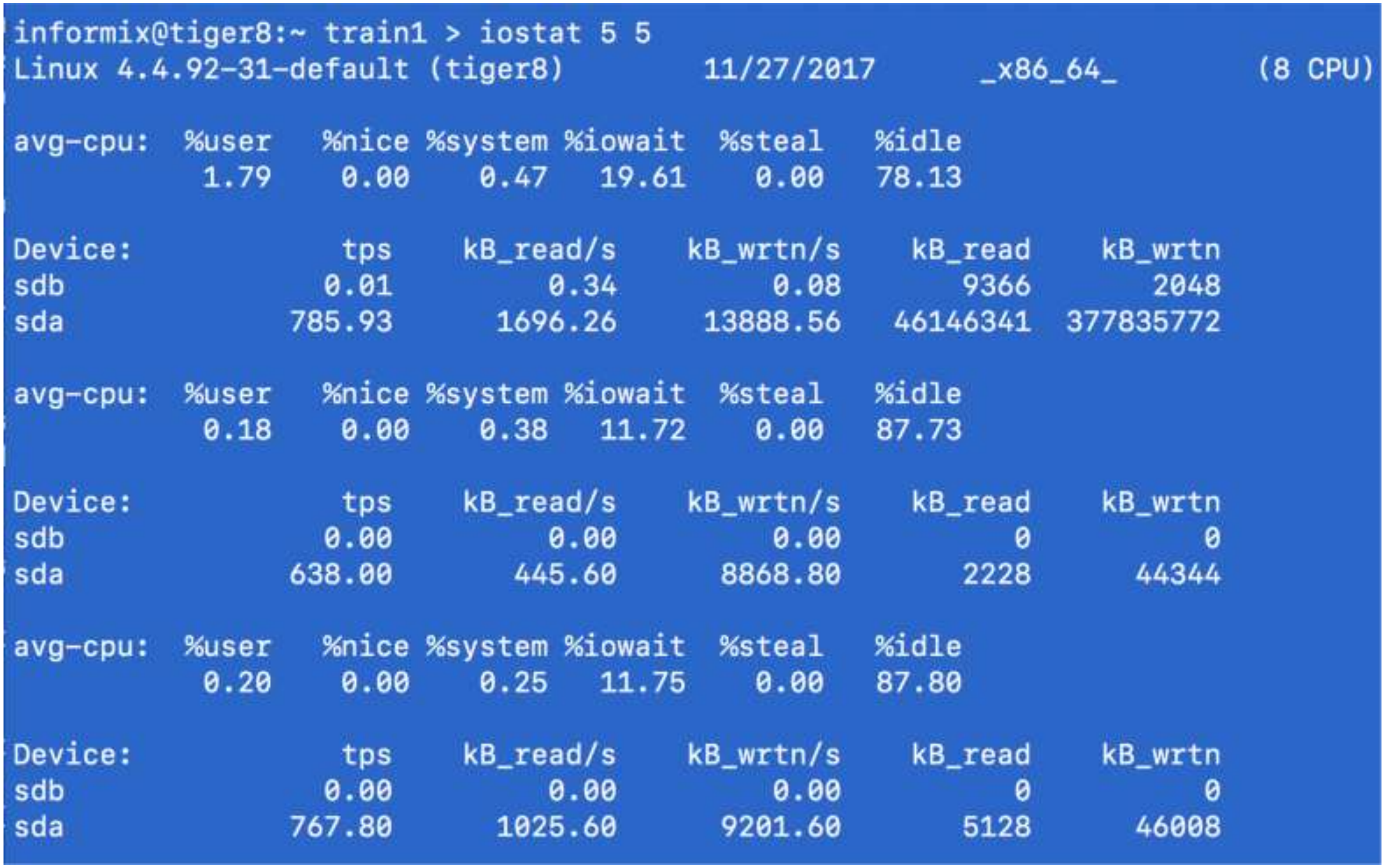
- Example on a 8 CPU machine:



```
informix@tiger8:~ train1 > mpstat -P ALL
Linux 4.4.92-31-default (tiger8)        11/27/2017      _x86_64_        (8 CPU)

06:47:07 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
06:47:07 PM  all    1.80    0.00    0.45   19.46    0.00    0.02    0.00    0.00    0.00   78.27
06:47:07 PM    0    1.57    0.00    0.34   15.27    0.00    0.02    0.00    0.00    0.00   82.80
06:47:07 PM    1    1.56    0.00    0.32   15.52    0.00    0.00    0.00    0.00    0.00   82.59
06:47:07 PM    2    0.84    0.00    0.45   15.76    0.00    0.09    0.00    0.00    0.00   82.86
06:47:07 PM    3    1.12    0.00    0.42   18.28    0.00    0.00    0.00    0.00    0.00   80.19
06:47:07 PM    4    2.44    0.00    0.58   26.51    0.00    0.00    0.00    0.00    0.00   70.47
06:47:07 PM    5    2.70    0.00    0.58   26.58    0.00    0.00    0.00    0.00    0.00   70.13
06:47:07 PM    6    3.16    0.00    0.51   21.15    0.00    0.00    0.00    0.00    0.00   75.18
06:47:07 PM    7    2.42    0.00    0.58   25.27    0.00    0.00    0.00    0.00    0.00   71.72
informix@tiger8:~ train1 > 
```

IOSTAT – Disk I/O Statistics



```
informix@tiger8:~ train1 > iostat 5 5
Linux 4.4.92-31-default (tiger8)        11/27/2017      _x86_64_        (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.79    0.00    0.47   19.61    0.00   78.13

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sdb               0.01         0.34         0.08       9366       2048
sda             785.93      1696.26     13888.56   46146341  377835772

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.18    0.00    0.38   11.72    0.00   87.73

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sdb               0.00         0.00         0.00          0          0
sda             638.00       445.60      8868.80       2228      44344

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.20    0.00    0.25   11.75    0.00   87.80

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sdb               0.00         0.00         0.00          0          0
sda             767.80      1025.60      9201.60       5128      46008
```

VXSTAT – Veritas Volume Manager Statistics

- Part of Veritas Volume Manager
- To display disk statistics, use the `vxstat -d` command:

```
             OPERATIONS          BLOCKS          AVG TIME(ms)
TYP NAME   READ     WRITE     READ     WRITE     READ  WRITE
dm  disk01  40473  174045    455898   951379    29.5  35.4
dm  disk02  32668  16873     470337   351351    35.2  102.9
dm  disk03  55249  60043     780779   731979    35.3  61.2
dm  disk04  11909  13745     114508   128605    25.0  30.7
```

PS – Unix Processes Statistics

- Key Options:

-e List information about every process now running.

-f Generate a full listing.

-l Generate a long listing.

-P Print the number of the processor to which the process or lwp is bound.

-t term List only process data associated with term.

-u uidlist List only process data whose effective user ID number or login name is given in uidlist.

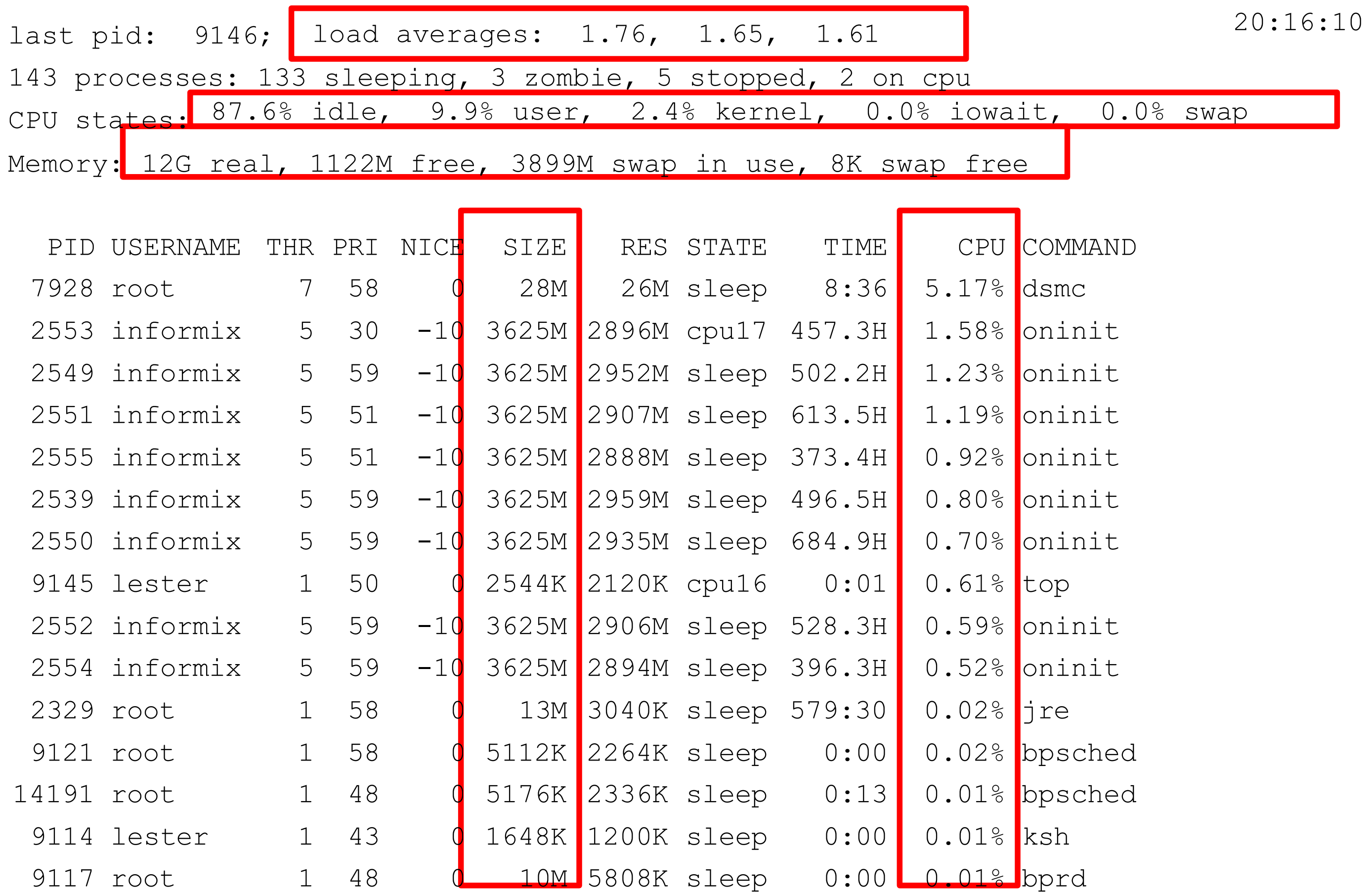
-U uidlist List information for processes whose real user ID numbers or login names are given in uidlist.

PS – Unix Processes Examples

```
lester@atlas >ps -fu informix | more
     UID   PID  PPID  C     STIME TTY       TIME CMD
informix   416     1  0    Apr 17 ?         0:05 oninit  -yv
informix   418   417  0    Apr 17 ?         0:05 oninit  -yv
informix   428     1  0    Apr 17 ?         0:11 oninit  -yv
informix  4085  3984  0  14:45:38 pts/2     0:00 dbaccess
informix  3984  3966  0  14:44:03 pts/2     0:00 bash
informix  3927     1  0  14:23:31 ?        16:21 oninit
informix  3966   874  0  14:37:34 pts/2     0:00 -ksh
```

```
lester@atlas >ps -lu informix | more
 F S   UID   PID   PPID  C PRI NI      ADDR     SZ     WCHAN TTY      TIME CMD
 8 S   202   416      1  0  41 20         ?  17648         ? ?        0:05 oninit
 c S   202   418    417  0  41 20         ?  17647         ? ?        0:05 oninit
 8 S   202   428      1  0  40 20         ?  14792         ? ?        0:11 oninit
 8 S   202  4085   3984  0  41 20         ?    654         ? pts/2    0:00 dbaccess
 8 S   202  3984   3966  0  51 20         ?    311         ? pts/2    0:00 bash
 8 S   202  3927      1  0  41 20         ?  17389         ? ?       16:21 oninit
 8 S   202  3966    874  0  51 20         ?    236         ? pts/2    0:00 ksh
```

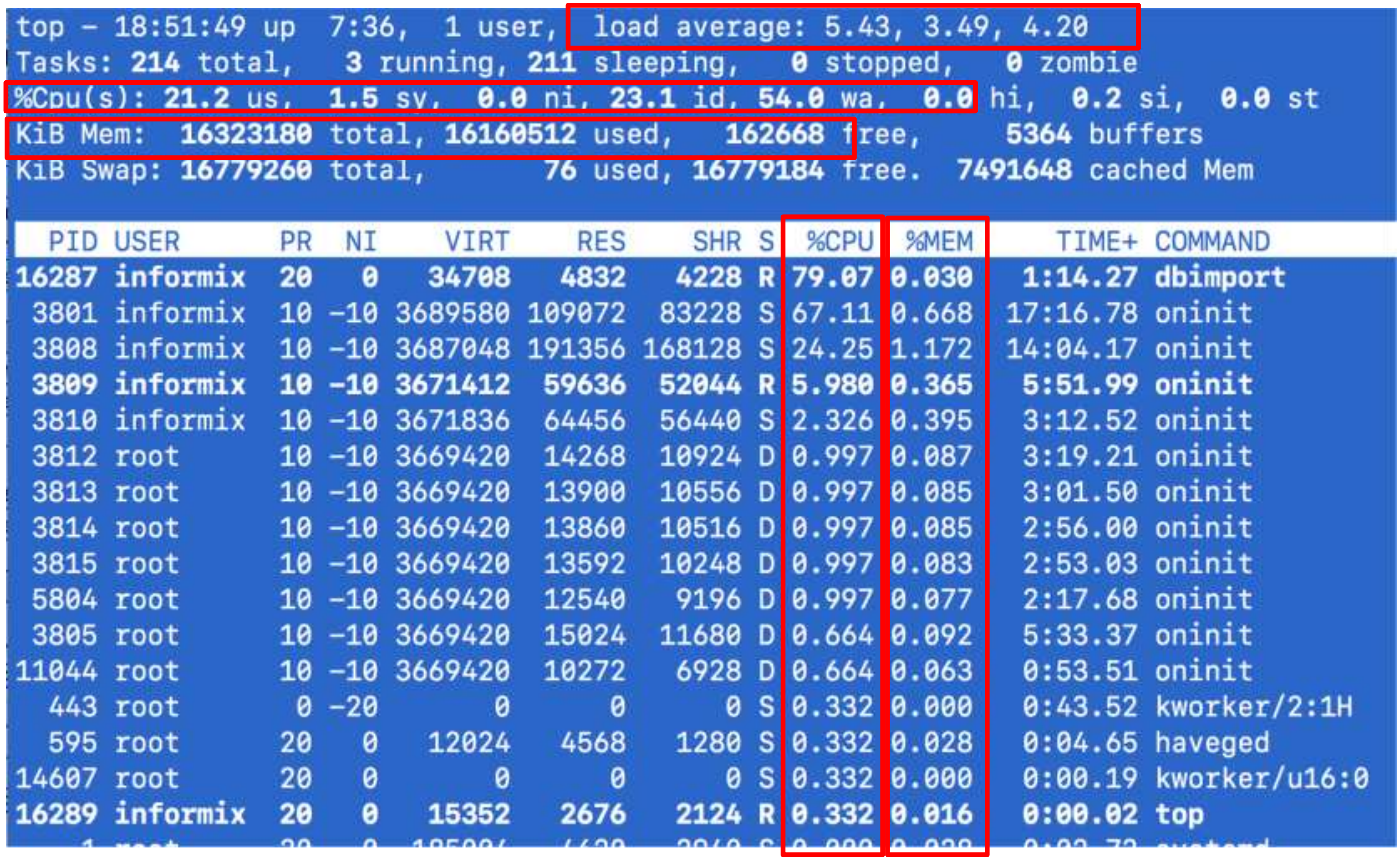
TOP – Top Unix Processes



```
last pid:  9146;  load averages:  1.76,  1.65,  1.61                    20:16:10
143 processes: 133 sleeping, 3 zombie, 5 stopped, 2 on cpu
CPU states: 87.6% idle,  9.9% user,  2.4% kernel,  0.0% iowait,  0.0% swap
Memory: 12G real, 1122M free, 3899M swap in use, 8K swap free

  PID USERNAME THR PRI NICE  SIZE   RES STATE   TIME    CPU COMMAND
 7928 root       7  58    0   28M   26M sleep   8:36  5.17% dsmc
 2553 informix   5  30  -10 3625M 2896M cpu17 457.3H  1.58% oninit
 2549 informix   5  59  -10 3625M 2952M sleep 502.2H  1.23% oninit
 2551 informix   5  51  -10 3625M 2907M sleep 613.5H  1.19% oninit
 2555 informix   5  51  -10 3625M 2888M sleep 373.4H  0.92% oninit
 2539 informix   5  59  -10 3625M 2959M sleep 496.5H  0.80% oninit
 2550 informix   5  59  -10 3625M 2935M sleep 684.9H  0.70% oninit
 9145 lester     1  50    0 2544K 2120K cpu16   0:01  0.61% top
 2552 informix   5  59  -10 3625M 2906M sleep 528.3H  0.59% oninit
 2554 informix   5  59  -10 3625M 2894M sleep 396.3H  0.52% oninit
 2329 root       1  58    0   13M 3040K sleep 579:30  0.02% jre
 9121 root       1  58    0 5112K 2264K sleep   0:00  0.02% bpsched
14191 root       1  48    0 5176K 2336K sleep   0:13  0.01% bpsched
 9114 lester     1  43    0 1648K 1200K sleep   0:00  0.01% ksh
 9117 root       1  48    0   10M 5808K sleep   0:00  0.01% bprd
```

TOP – Top Unix Processes



```
top - 18:51:49 up  7:36,  1 user,  load average: 5.43, 3.49, 4.20
Tasks: 214 total,   3 running, 211 sleeping,   0 stopped,   0 zombie
%Cpu(s): 21.2 us,  1.5 sy,  0.0 ni, 23.1 id, 54.0 wa,  0.0 hi,  0.2 si,  0.0 st
KiB Mem:  16323180 total, 16160512 used,   162668 free,     5364 buffers
KiB Swap: 16779260 total,       76 used, 16779184 free.  7491648 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
16287 informix  20   0   34708   4832   4228 R 79.07 0.030   1:14.27 dbimport
 3801 informix  10 -10 3689580 109072  83228 S 67.11 0.668  17:16.78 oninit
 3808 informix  10 -10 3687048 191356 168128 S 24.25 1.172  14:04.17 oninit
 3809 informix  10 -10 3671412  59636  52044 R 5.980 0.365   5:51.99 oninit
 3810 informix  10 -10 3671836  64456  56440 S 2.326 0.395   3:12.52 oninit
 3812 root      10 -10 3669420  14268  10924 D 0.997 0.087   3:19.21 oninit
 3813 root      10 -10 3669420  13900  10556 D 0.997 0.085   3:01.50 oninit
 3814 root      10 -10 3669420  13860  10516 D 0.997 0.085   2:56.00 oninit
 3815 root      10 -10 3669420  13592  10248 D 0.997 0.083   2:53.03 oninit
 5804 root      10 -10 3669420  12540   9196 D 0.997 0.077   2:17.68 oninit
 3805 root      10 -10 3669420  15024  11680 D 0.664 0.092   5:33.37 oninit
11044 root      10 -10 3669420  10272   6928 D 0.664 0.063   0:53.51 oninit
  443 root       0 -20       0      0      0 S 0.332 0.000   0:43.52 kworker/2:1H
  595 root      20   0   12024   4568   1280 S 0.332 0.028   0:04.65 haveged
14607 root      20   0       0      0      0 S 0.332 0.000   0:00.19 kworker/u16:0
16289 informix  20   0   15352   2676   2124 R 0.332 0.016   0:00.02 top
```

NETSTAT – Network Statistics

- Options:

```
usage: netstat [-anv] [-f address_family]
       netstat [-g | -p | -s] [-n] [-f address_family] [-P protocol]
       netstat -m
       netstat -i [-I interface] [-an] [-f address_family] [interval]
       netstat -r [-anv] [-f address_family]
       netstat -M [-ns] [-f address_family]
       netstat -D [-I interface] [-f address_family]
```

- Example

```
lester@atlas >netstat -i
Name  Mtu  Net/Dest         Address          Ipkts  Ierrs Opkts  Oerrs Collis Queue
hme0  1500 atlas.addt.com atlas.addt.com 92751   0     50571   0     0      0
lo0   8232 loopback         localhost        80430  0     80430   0     0      0
```

Performance Metrics Goals

CPU Monitoring

- Are the CPUs overloaded?
- Factors:
  - Number of CPUs
  - Speed of CPUs (old vs new systems)
  - Number of process needing CPU time.

How Busy are the CPU’s?

- Tools to monitor:
  - sar –u
  - vmstat
  - mpstat
  - top, prstat
- **Performance Guideline - % CPU busy:**
  - < 30 % - Good
  - 30-60% - Fair
  - > 60% - Poor

SAR – Example

```
00:00:00     %usr    %sys    %wio   %idle
07:00:00     27         3       0      70
07:15:02     61         6       0      33  Poor
07:30:01     47         4       0      49
07:45:01     28         3       0      70
08:00:00     30         2       0      68
                                          Fair
08:15:00     50         3       0      46
08:30:01     56         3       0      41
08:45:00     22         2       0      77  Good
```

How many process are waiting to run on the CPUs?

- Tools to monitor Load Average:
  - sar –q
  - Uptime
- **<u>Performance Guideline – number of waiting processes:</u>**
  - < 2 per CPU – Good
  - 2-4 per CPU – Fair
  - > 4 per CPU – Poor

CPU Load Average Example:

```
lester@atlas >uptime
  9:58pm  up 2 day(s),  5:52,  4 users,  load average: 0.03, 0.04,
    0.04
```

- Displays run queue over the last 1, 5, and 15 minutes
- On a 4 CPU machine:
  - < 2 x 4 = Good
  - 2-4 x 4 = Fair
  - > 4 x 4 = Poor

Memory Monitoring

- Is memory being over-used or under-used?
- Memory shortage causing swapping to disk.
- Factors:
  - Amount of RAM
  - 32 bit vs 64 bit OS and applications
  - 32 bit Informix IDS limited to:
    - 3.6 GB on Solaris
    - 2 GB on AIX
    - 2 GB on Windows
- One of best Informix IDS performance improvements is adding BUFFERS

How much Memory is Used?

- Tools to monitor
  - top
  - sar –r
  - vmstat
- Performance Guidelines
  - Don’t monitor free memory since a good OS will use all extra memory as file system cache
  - Monitor swap space and paging in/outs

Memory – Key is to Monitor Paging In/Out

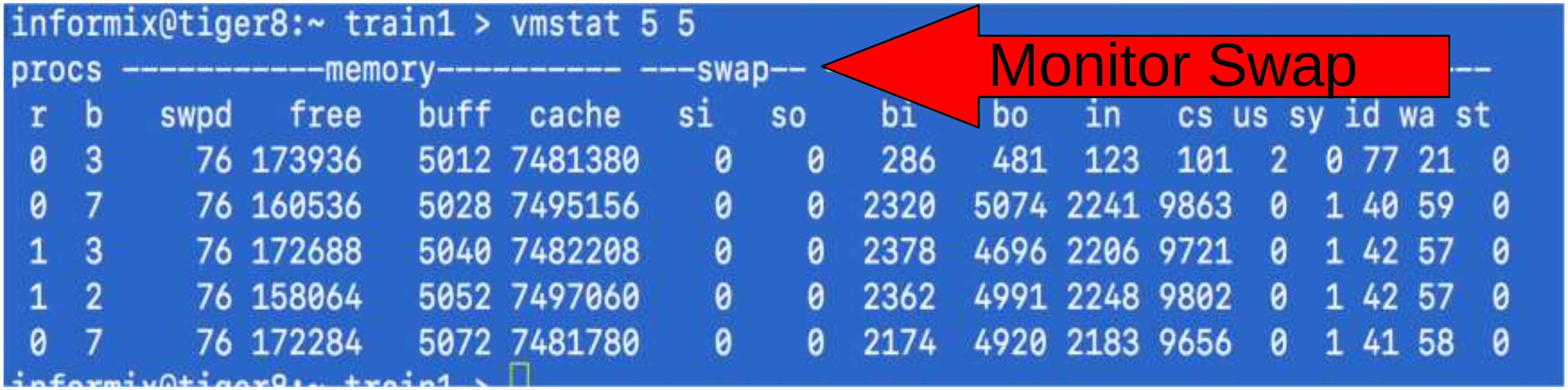
- Monitor vmstat:
  - pi - kilobytes paged in
  - po - kilobytes paged out

Out of Memory

- Monitor sar –g
  - pgout/s - page-out requests per second.
  - ppgout/s - pages paged-out per second.
- Monitor sar –p
  - pgin/s - page-in requests per second.
  - ppgin/s - pages paged-in per second.

Memory – Key is to Monitor Swap In/Out

- Monitor vmstat:



```
informix@tiger8:~ train1 > vmstat 5 5
procs -----------memory---------- ---swap-- -                                  ---
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  3     76 173936   5012 7481380    0    0   286   481  123  101  2  0 77 21  0
 0  7     76 160536   5028 7495156    0    0  2320  5074 2241 9863  0  1 40 59  0
 1  3     76 172688   5040 7482208    0    0  2378  4696 2206 9721  0  1 42 57  0
 1  2     76 158064   5052 7497060    0    0  2362  4991 2248 9802  0  1 42 57  0
 0  7     76 172284   5072 7481780    0    0  2174  4920 2183 9656  0  1 41 58  0
informix@tiger8:~ train1 >
```

Monitor Swap

What Processes are Using the Most Memory?

- Tools to monitor – look at the SIZE column:
  - top
  - prstat
  - ps
- Performance Guideline for Informix:
  - BUFFERS - number of shared memory buffers
  - SHMVIRTSIZE - initial virtual shared memory segment size
  - SHMADD - size of new shared memory segments
  - SHMTOTAL – total size of shared memory

Monitoring Disks

- Goal is to balance I/O across all disks
  - Use: sar and iostat
- Find the FAST spot on the disk and locate key chunks there
- Find the optimal disk throughput

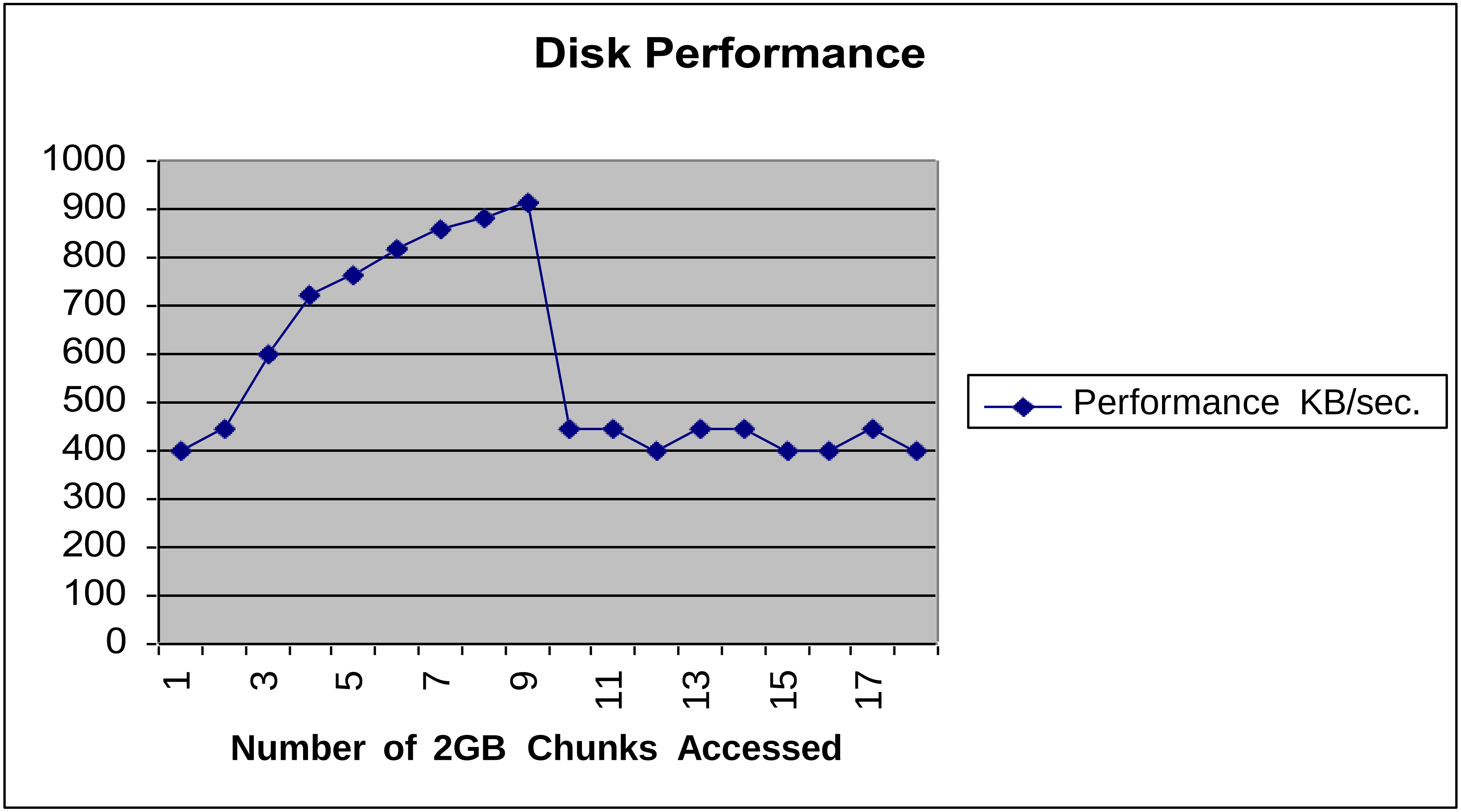
Disk Throughput

- Example pfread – 2 GB chunks on a 72 GB disk

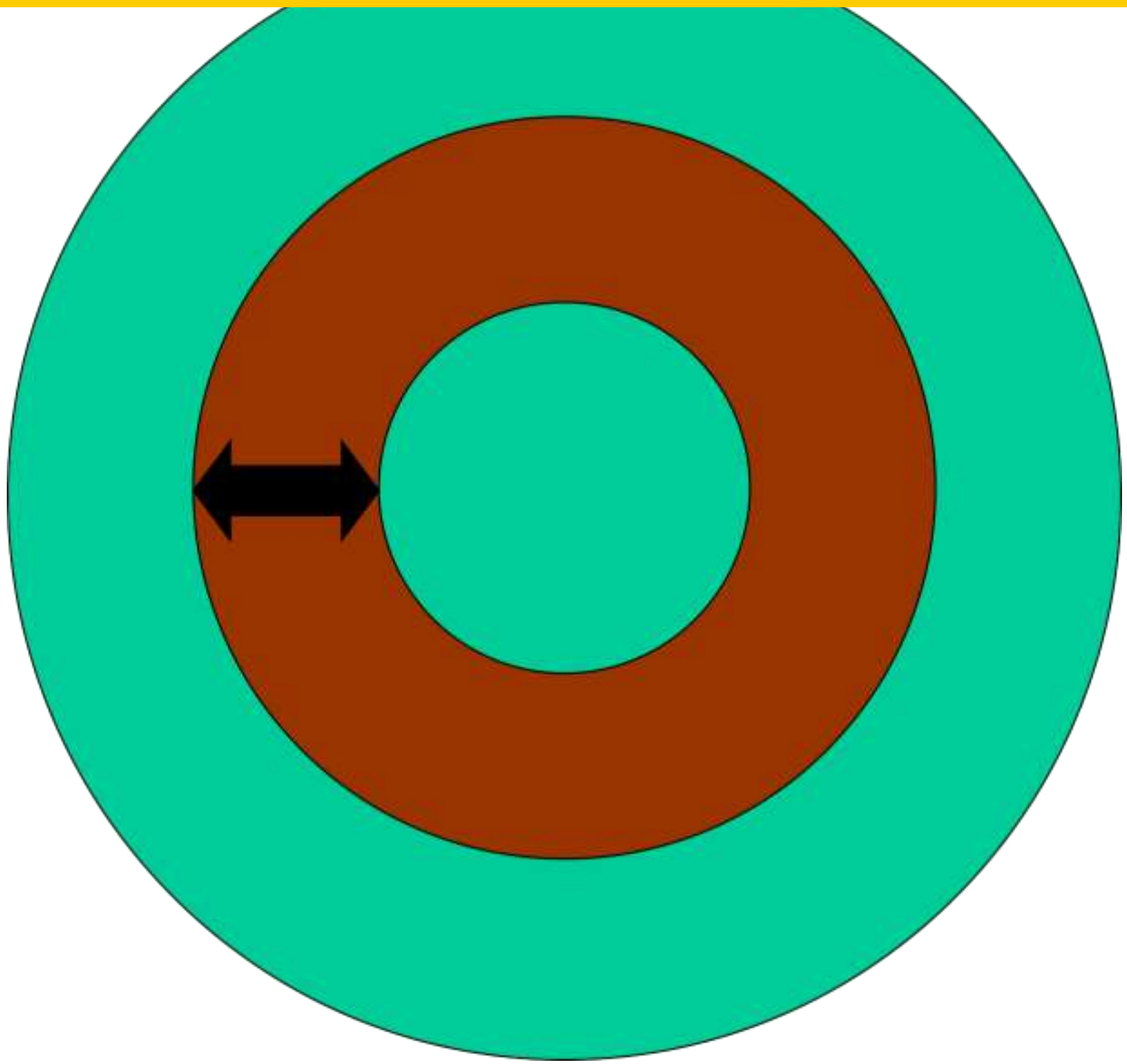
```
pfread.ksh 1 30 /informixchunks/d4chk14
 /informixchunks/d4chk14  :        1 concurrent  read  threads          500  KB/sec.
 /informixchunks/d4chk14  :        2 concurrent  read  threads          500  KB/sec.
 /informixchunks/d4chk14  :        3 concurrent  read  threads          750  KB/sec.
 /informixchunks/d4chk14  :        4 concurrent  read  threads          800  KB/sec.
 /informixchunks/d4chk14  :        5 concurrent  read  threads          1000  KB/sec.
 /informixchunks/d4chk14  :        6 concurrent  read  threads          996 KB/sec.
 /informixchunks/d4chk14  :        7 concurrent  read  threads          1071  KB/sec.
 /informixchunks/d4chk14  :        8 concurrent  read  threads          1082  KB/sec.
 /informixchunks/d4chk14  :        9 concurrent  read  threads          1125  KB/sec.
 /informixchunks/d4chk14  :        10  concurrent  read  threads        500  KB/sec.
 /informixchunks/d4chk14  :        11  concurrent  read  threads        444  KB/sec.
 /informixchunks/d4chk14  :        12  concurrent  read  threads        500  KB/sec.
```

- Best performance is using 9 x 2GB chunks = 18GB of the 72 GB disk

Disk Throughput – 36 GB Disk



Disk Layout - The FASTEST location on a disk is where the disk arm has to move the least to read or write data

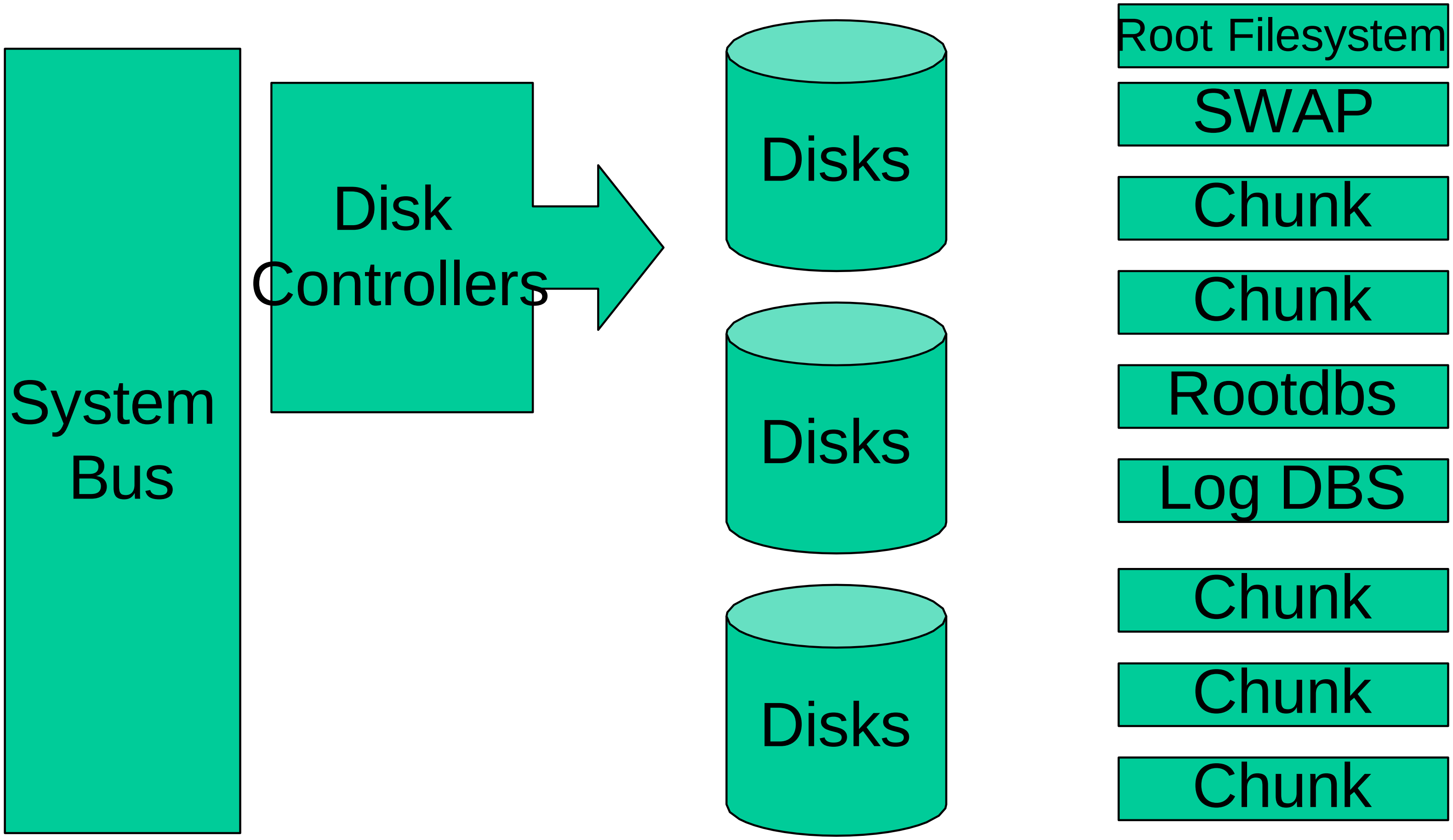


Monitor Disk I/O with SAR

- Report activity for each block device (disk or tape)
  - %busy – portion of time device was busy servicing a transfer request – How busy are your disks?
  - avque – average number of requests outstanding during that time.
  - read/s, write/s, blks/s - number of read/write transfers from or to device, number of bytes transferred in 512-byte units.
  - avwait - average wait time in milliseconds.
  - avserv - average service time in milliseconds.
- Example sar –d

```
00:00:00    device         %busy    avque   r+w/s   blks/s   avwait  avserv
00:15:00    nfs1               0      0.0       0        0      0.0     0.0
            sd7               11      0.7      17      225      0.0    40.2
            sd7,a              0      0.0       0        0      0.0     0.0
            sd7,b              0      0.0       0        0      0.0     0.0
            sd7,c              0      0.0       0        0      0.0     0.0
            sd7,d              0      0.0       0        0      0.0     0.0
            sd7,e             11      0.7      17      225      0.0    40.2
```

Map Your Disk Drives



Create a Disk Layout Spreadsheet

- Controller/ Disk Array
- Disk
- Logical Volumes or Slices
- Chunks, Filesystems, etc…
- Tables in Chunks
- Compare results from sar -d and onstat -d

Disk Performance Spreadsheet

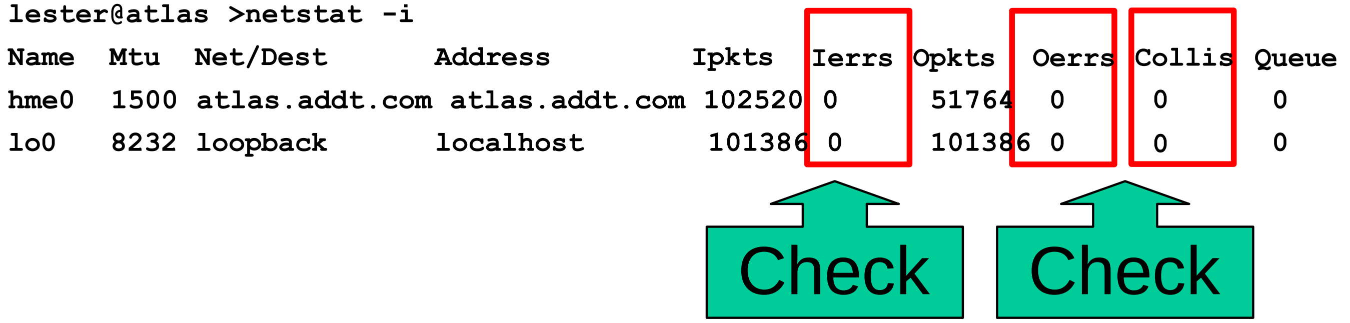
Disk Performance					
Disk Layout				Performance	
Controller	Disk	Volume	Chunk/Filesystem	onstat -d	sar -d
c1	disk1	d1v1			
c1	disk1	d1v2			
c1	disk1	d1v3			
c1	disk1	d1v4			
c1	disk2	d2v1			
c1	disk2	d2v2			

Monitoring Network

- How measure real output of network interface?
  - FTP Test – How long does it take to ftp a 2GB file to your destination? KB per second
  - Database server cannot send data out any faster than ftp
- Measure network errors and collusions
  - Netstat –i

Network Errors and Collisions

- Tool to monitor:
  - netstat –i
- Example output:



```
lester@atlas >netstat -i
Name  Mtu  Net/Dest       Address        Ipkts  Ierrs Opkts  Oerrs Collis  Queue
hme0  1500 atlas.addt.com atlas.addt.com 102520  0     51764  0      0      0
lo0   8232 loopback       localhost       101386  0    101386 0      0      0
```

Check

Check

- Performance Guideline – no errors or collisions